University Politehnica of Bucharest
Faculty of Automatic Control and Computers
Computer Science and Engineering Department

Diploma Thesis

Mobile Gateway for Wireless Sensor Networks utilizing drones

by

Ioan Deaconu

Supervisor: Prof. Dr. Ing. Nicolae Ţăpuş
Supervisor: As. Drd. Ing. Andrei Voinescu

Bucharest, September 2014

Contents

Contents i

1 Introduction 3

2 Related Work 4

3 Hardware Platform 5
- 3.1 The Parrot AR.Drone 2.0 . . . 5
- 3.2 The Sparrow Family . . . 6
  - 3.2.1 The Sparrow Dongle . . . 7
  - 3.2.2 The SparrowV32 . . . 8

4 Software Implementation 9
- 4.1 The Debug Module . . . 10
- 4.2 The Data Collecting Module . . . 10
  - 4.2.1 Modules intercommunication . . . 11
  - 4.2.2 Failure proof . . . 11
- 4.3 The Communication Module . . . 11
  - 4.3.1 Socket with connection reset . . . 12
  - 4.3.2 JSON Encoding of Data . . . 12
- 4.4 Android application modules . . . 12
  - 4.4.1 Display information module . . . 13
  - 4.4.2 FTP communication module . . . 14

5 Conclusions 15
- 5.1 Outlook . . . 15

Bibliography 16

List of Figures 18

List of Tables **19**

Listings **20**

Abstract

Keywords Wireless Sensor Networks, task, scheduling, graph cuts

Acknowledgements

Chapter 1

Introduction

Thesis Intro - no more than 3 pages.

Chapter 2

Related Work

Related work

Chapter 3

Hardware Platform

In this chapter we will present the hardware platforms used.

Because we wanted to emphasise not only a new way of aquiring data, but also a simple and low cost one, we have selected The Parrot AR.Drone 2.0 as the work horse that will carry the gateway that will communicate with the nodes from our Sparrow Family .

chesti ..... help .. anybody

3.1 The Parrot AR.Drone 2.0

Parrot AR.Drone is a wifi radio controlled flying quadcopter built by the French company Parrot. The original drone was released in 2010 and in 2012 it was replaced by version 2.0. Since the launch of the original AR.Drone, more the half a milion units have been sold, making it one of the, if not, the most popular drone on the market.

The reason of its success is not entirely due to the relatively low price of around 300$ but also because it is very easy to learn how to control the drone and also because of the usb port that accomodate any device using that interface and the linux operating system

Because of those reasons, the Drone has a number of aftermaket modules that can be attached to it like the Flight Recorder GPS Module. This module has a built in storage of 4GB for video recording purposes and a built in GPS receiver. This allows the drone to follow a predetermined path of waypoints and to return back from where it took off automatically, all within the limit of the Wi-Fi connection with the control device.

The arrot AR.Drone 2.0 specifications are :

- 1GHz 32 bit ARM Cortex A8 processor with 800MHz video DSP TMS320DMC64x

Figure 3.1: *The arrot AR.Drone 2.0*

- Linux 2.6.32
- 1Gbit DDR2 RAM at 200MHz
- USB 2.0 high speed for extensions
- Wi-Fi b,g,n
- 3 axis gyroscope 2000/second precision
- 3 axis accelerometer +-50mg precision
- 3 axis magnetometer 6 precision
- Pressure sensor +/- 10 Pa precision
- Ultrasound sensors for ground altitude measurement
- 60 fps vertical QVGA camera for ground speed measurement
- 30 fps 720p front mounted camera

3.2 The Sparrow Family

The Sparrow Family , composed of the Sparrow Dongle and SparrowV32, use a 2,4 GHz wireless network as a medium of communication.

At heart,soul and brain of this family lies the ATMega128RFA1. It is an 8bit microcontroller from Atmel that has a very important on-chip 2.4 GHz wireless transceiver that helps in keeping the footprint of the board very small.

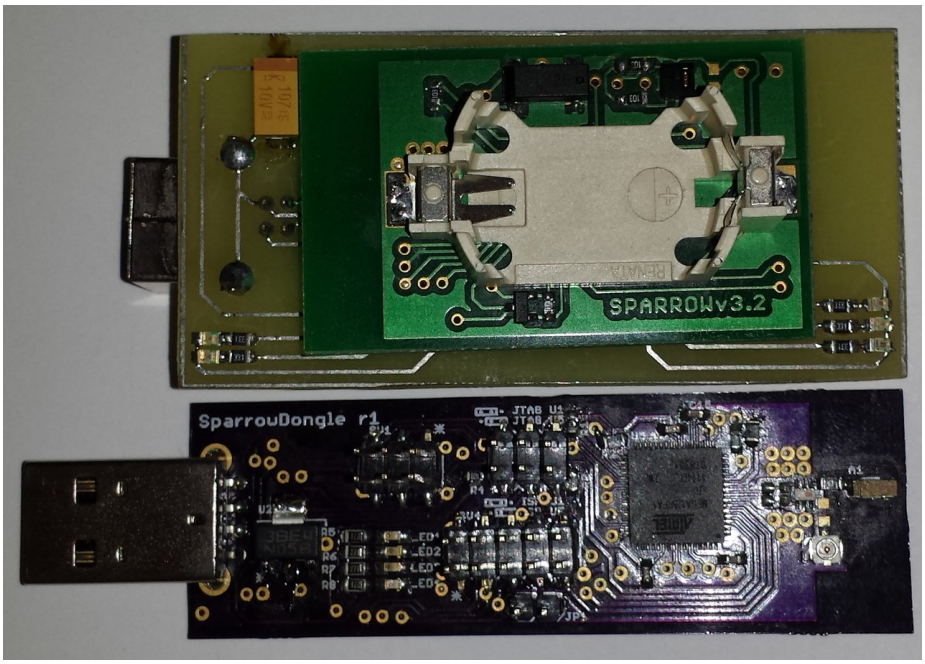


Figure 3.2: *The Sparrow Dongle next to the SparrowV32*

The signal received or sent by the wireless transceiver can be boosted by attaching an external atena. For example, in an ideal situation, with no interferences from the outside world , an 8-dbi omni-directional antenna mounted on both communication devices would amount to an around 200 meters of communication range, well over the 70 meters measured with the default antennas.

3.2.1 The Sparrow Dongle

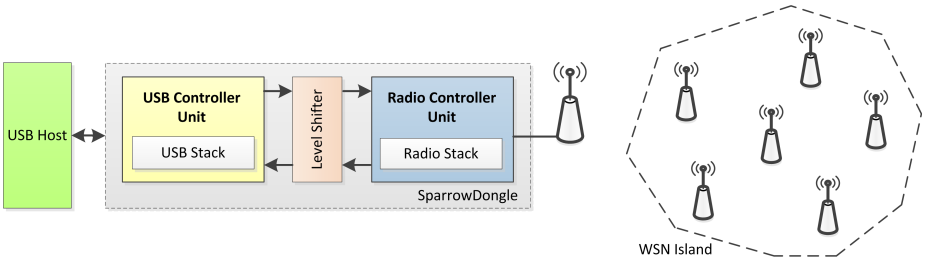


Figure 3.3: *SparrowDongle stick architecture*

The unseen hero of the WSN world, the Sparrow Dongle is the gateway of the Sparrow Family. The Dongle can be connected to any devices that has an

usb port and can support USB CDC with ACM module (USB Communications Device Class with Abstract Control Mode).

The dongle uses Atmega32U4 as a dedicated USB Controller unit. This design allows the RF controller to run any RF communication stack without having the USB code intrude on key timings.

3.2.2 The SparrowV32

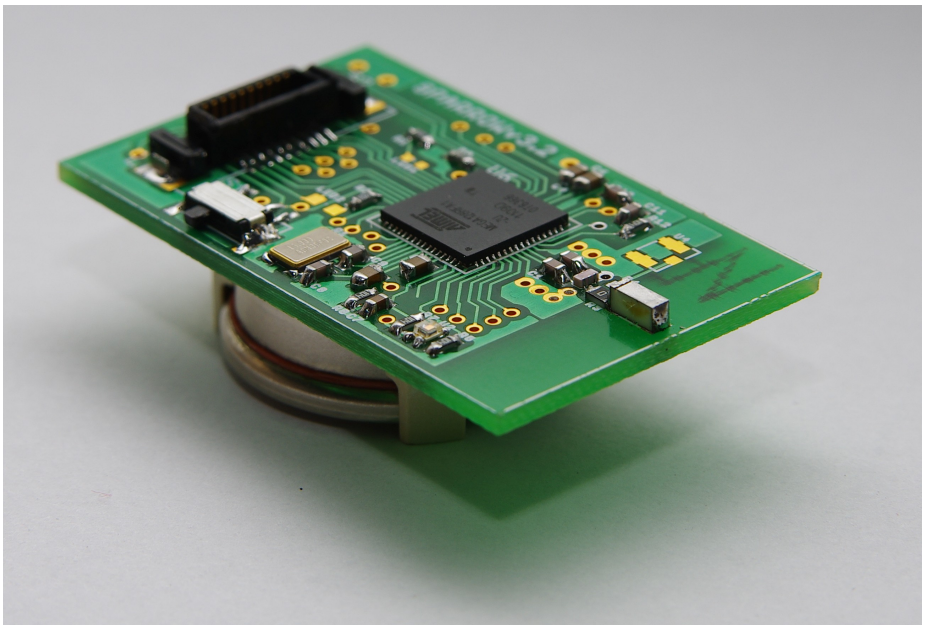


Figure 3.4: *The SparrowV32*

The main hero of The Sparrow Family, the SparrowV32 nodes does all the dirty work of collecting data from the surrounding environment and send it to the Sparrow Dongle. The collected data depends on the sensors of the SparrowV32. The standard implementation has a light, temperature and humidity sensor. Besides this information, it also sends the battery level.

Chapter 4

Software Implementation

The entire solution is devided into different modules that run independently but communicate with each other to achive the main purpose. The idea of software organized as modules allows for future features to be added more easily.

The main modules are installed in the AR Parrot Drone and the android FreeFlight 2.0 application.

The SparrowDongle gateway is always in a listen for data state and dumps every data received on the serial. When it receives the data, it sends back an ack to let the SparrowV3.2 node to know that it can begin sending the entire stored data to the mobile gateway.

The SparrowV3.2 node is sending periodically a small data packet to check if a gateway is available. When it receives the ack for the packet it starts sending the stored data to the gateway. The data sent can vary, from sensor readings to debugging informations in order to check the state of the Wireless Sensor Network.

The data gathered by the gateway is saved into different files in the AR Parrot Drone's internal memory. The files also contains informations such as the node identification tag and time of the transfer. The data can be accessed at any time by any device connected to the drone's wireless network port 4242 via FTP.

The drone also process some collected data to provide informations like signal strength, last connection time and number of nodes. This informations are send to the controlling device through a socket connection.

The controling device, pc or android, will gather the informations and display them to the user.

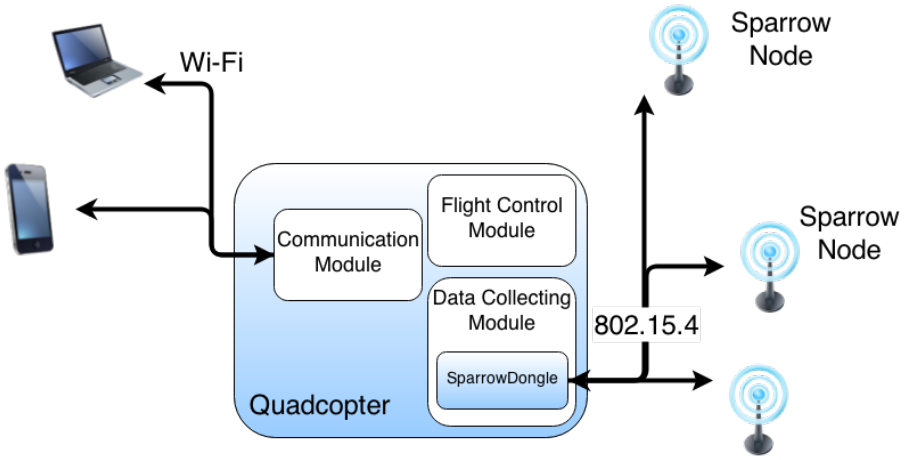


Figure 4.1: *Modules and connections between them and devices*

4.1 The Debug Module

The code can be compiled to display debug informations to the console or to supress them entirely. Also, in certain parts of the modules, a wait action is need in order to wait for an action to be executed. Now the delay is set at 100 ms, but it can be easily modiffied to any desired value.

Listing 4.1: Data Collection use of mutex

```
/* activates/deactivates printf debuf information*/
#define DEBUG_ON 0
/* delay yime in microseconds*/
#define DELAY_US 100000
#define DEBUG_PRINT(a...) { if(DEBUG_ON) printf(a); }
```

4.2 The Data Collecting Module

The module saves the collected data int-o the drones internal memory and pases the data in order to gather certain informations like number of nodes currently connected to the Dongle, de signal strength, if the Dongle is connected etc. This informations are passed to the communication module to provide to the user realtime feedback.

This module, besides the main purpose and similar to the other modules, has some extra features that are design to make the solution more user friendly and easier to improve in the feature.

4.2.1 Modules intercommunication

The memory area in which the informations sent to the user are saved is shared between this module and the comunication module. Basically, the way this two modules interract with each other can be compared to the consumer - producer problem, where the Data Collecting Module can be associated with the producer side and the Communication Module with the consumer side.

The main problem consists of deadlocks and data starvation. This is prevent with the use of one mutex that allows only one thread at a time to moddify the informations.

Listing 4.2: Data Collection use of mutex

```
pthread_mutex_lock(&data_lock);
add_node_data(get_current_timestamp(),read_data + 7);
pthread_mutex_unlock(&data_lock);
```

The mutex is used similarly in the Communication Module when it consumes the information.

4.2.2 Failure proof

Because the Dongle is connected to an USB port on a machine that has a lot of vibrations, it might desconnect / reconnect for a very short period of time, so this module has been designed with multiple USB disconnects and reconnects without the need to reset the Drone. This infomation is vital, because you can check if the Dongle is still connected to the dron without the need to inspect it visualy or to connect to a debug terminal.

4.3 The Communication Module

All the information gathered by the Data Collecting Module would be useless if it cannot be accesed easily.

This module, as the name suggests, handles the communication of this this crucial information back to the user.

Being a different module, with different attributions than the Data Collecting Module, it has an entire proccess dedicated to it for 3 important reasons:

- 1. This approach of a module with its one procces allows the modules to run indepently of each other;
- 2. The Data Collecting Module can collect the data from the Dongle as soon as as it has a new one available;
- 3. If the Communication Module stops working, the Data Collecting Module continues to save the new informations received.

4.3.1 Socket with connection reset

The communication is done through socket connections listening on port 8888. It accepts only one connection at a time.

If a connected client decides to disconnect before or while a write is performed, a SIGPIPE error signal will be thrown, stopping all the modules. This is prevent by ignoring the signal, forcing the write acction to return a EPIPE, an exiting the this function.

The main proccess will calback the accept_socket_connection to reestablish a new connetion. Once a connection is established, it will send information once every DELAY_US microseconds. The program was configured and tested with a 100 ms wait period that leeds to a ten times per second information update.

This delay is required because:

- If data is send to often , the socket might be flooded and stop sending the data
- If there were no delay, it will create an big and useless processor busy state both for the drone and the controlling device.

4.3.2 JSON Encoding of Data

The data sent is encoded in JSON format, because it is very easy to encode and all devices can decode it.

The informations ecoded are

- Dongle conection status
- An array containing node informations
  - Node unique ID
  - Last conection time of the node to Dongle
  - The power of the received signal

4.4 Android application modules

Being an opensource platform we have moddified the AR Freeflight 2.0 android application to communication with our new modules added tot the drone.

Android fairly imposes the use of the background process class AsynkTask when you have to use communication protocols like http, ftp, sockets. This prevents the ui process from being stuck in communication and not responding to user actions.

The class offers 5 very important methods that can be overwritten, 3 running on the main ui process, that prepare data before and after exection, publish the

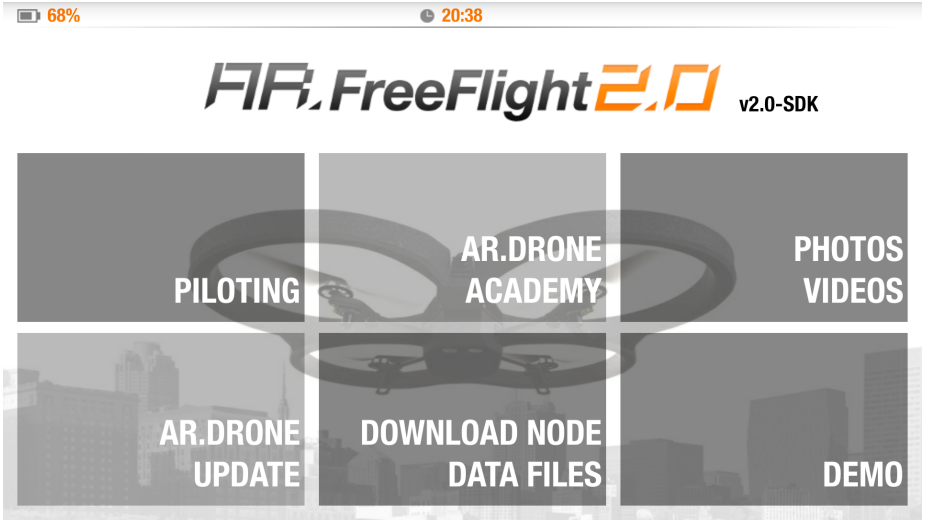


Figure 4.2: *ARFreeflight modified application*

progress or simply cancel at any step, and 1 running on the actual background process.

4.4.1 Display information module

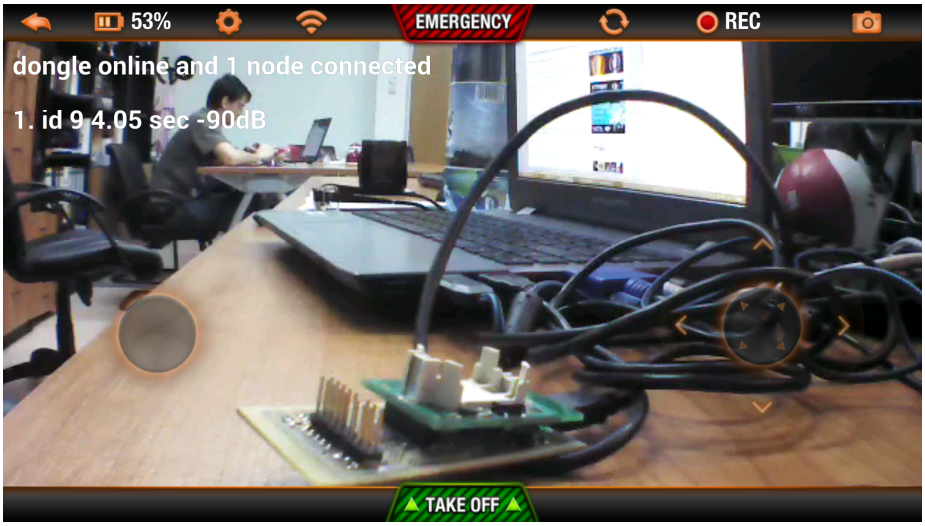
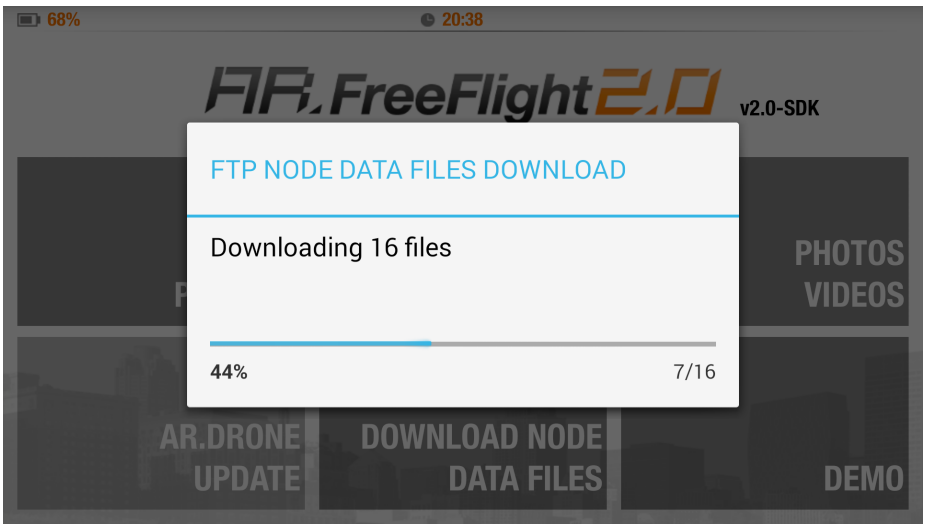


Figure 4.3: *ARFreeflight modified Piloting Screen*

The Piloting screen of the application has been modified to the display the received informations from the drone.

The information displayed consists of the dongle being functional and at most, 9 nodes sorted descending after their signal strength.

4.4.2 FTP communication module



W

Figure 4.4: *ARFreeflight FTP downloading files*

The drone has a build in FTP server that can be configured to allow acces to any folders/files on the drone. We have configured the drone so that the folder that contains the saved data can be accesed at any time using the 4242 port by any device that has FTP client capabilities. We have added this feature to the android application as well.

The application will download all the files from the drone to the local storage of your android device while displaying the progress.

Chapter 5

Conclusions

5.1 Outlook

Bibliography

[Atm] Atmel. Rzraven hardware user's guide. [cited at p. -]

[Cal04] Edgar H. Callaway. *Wireless Sensor Networks: Architectures and Protocols.* CRC Press, 2004. [cited at p. -]

[CK03] C. Chong and S.P. Kumar. In *Sensor networks: Evolution, opportunities, and challenges*, 2003. [cited at p. -]

[DPdR+05] Flvia Coimbra Delicato, Fbio Protti, Jos Ferreira de Rezende, Luiz F. Rust da Costa Carmo, and Luci Pirmez. Application-driven node management in multihop wireless sensor networks. In Pascal Lorenz and Petre Dini, editors, *ICN (1)*, volume 3420 of *Lecture Notes in Computer Science*, pages 569–576. Springer, 2005. [cited at p. -]

[ea07] Roberto Verdone et al. *Wireless Sensor and Actuator Networks: Technologies, Analysis and Design.* Academic Press, 1st edition, 2007. [cited at p. -]

[GH88] Olivier Goldschmidt and Dorit S. Hochbaum. Polynomial algorithm for the k-cut problem. 1988. [cited at p. -]

[HJ05] Tarek Hagras and Jan Janecek. A high performance, low complexity algorithm for compile-time task scheduling in heterogeneous systems. *Parallel Computing*, 31(7):653–670, 2005. [cited at p. -]

[Jac06] Brian Jacokes. Lecture notes on multiway cuts and k-cuts, July 2006. [cited at p. -]

[KD06] Snehal Kamalapur and Neeta Deshpande. Efficient cpu scheduling: A genetic algorithm based approach. *Ad Hoc and Ubiquitous Computing*, pages 206 – 207, December 2006. [cited at p. -]

[Li08] Xiang-Yang Li. *Wireless Ad Hoc and Sensors Networks: Theory and Applications.* Cambridge University Press, 2008. [cited at p. -]

[NGT99] Andrew Goldberg Nec, Andrew V. Goldberg, and Kostas Tsioutsiouliklis. Cut tree algorithms. In *In Symposium on Discrete Algorithms*, pages 376–385, 1999. [cited at p. -]

[PB05] C. Prehofer and C. Bettstetter. Self-organization in communication networks: principles and design paradigms. *IEEE Communications Magazine*, 43(7):78–85, July 2005. [cited at p. -]

[TEÖ07] Yuan Tian, Eylem Ekici, and Füsun Özgüner. Energy-constrained task mapping and scheduling in wireless sensor networks. 2007. [cited at p. -]

[YJK07] William M Iversen Yunseop (James) Kim, Robert G Evans. The future of intelligent agriculture. *Resource*, October 2007. [cited at p. -]

[zig] Zigbit wireless modules datasheet. [cited at p. -]

List of Figures

3.1 *The arrot AR.Drone 2.0* 6
3.2 *The Sparrow Dongle next to the SparrowV32* 7
3.3 *SparrowDongle stick architecture* 7
3.4 *The SparrowV32* 8

4.1 *Modules and connections between them and devices* 10
4.2 *ARFreeflight modified application* 13
4.3 *ARFreeflight modified Piloting Screen* 13
4.4 *ARFreeflight FTP downloading files* 14

List of Tables

Listings

4.1 Data Collection use of mutex 10
4.2 Data Collection use of mutex 11